Sophisticated Media Playback and Animation Through Open Web Standards

Session 205

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

Prerequisite Download Required for This Session

To be ready for this session:

- Have Safari 4.0 installed
- Have Xcode running
- Download sample code

Materials available at:
http://developer.apple.com/wwdc/attendee/

Sophisticated Media Playback and Animation Through Open Web Standards

Hands-on

Antoine Quint
Software Engineer
Eric Carlson
Senior Software Engineer

Goal of This Session

Practical integration of key technologies

- HTML 5 <video> and <audio> elements—media in your Web page
- HTML 5 Media Events—tracking media state
- HTML 5 Media APIs—controlling media

Putting <video> in Context

The New York Times

flickr

YouTube

hulu

PANDORA

Pitchfork.tv

vimeo

Dailymotion

last·fm

“As of today, 20 hours of video are uploaded to YouTube every minute.”

Ryan Junee
YouTube

That’s about 3.3 years worth of content… every single day!

Why <video>?

Just another element

- Easier to author—like `<img>`
- Integrates with Web technologies—CSS, DOM, etc.

Optimized integration

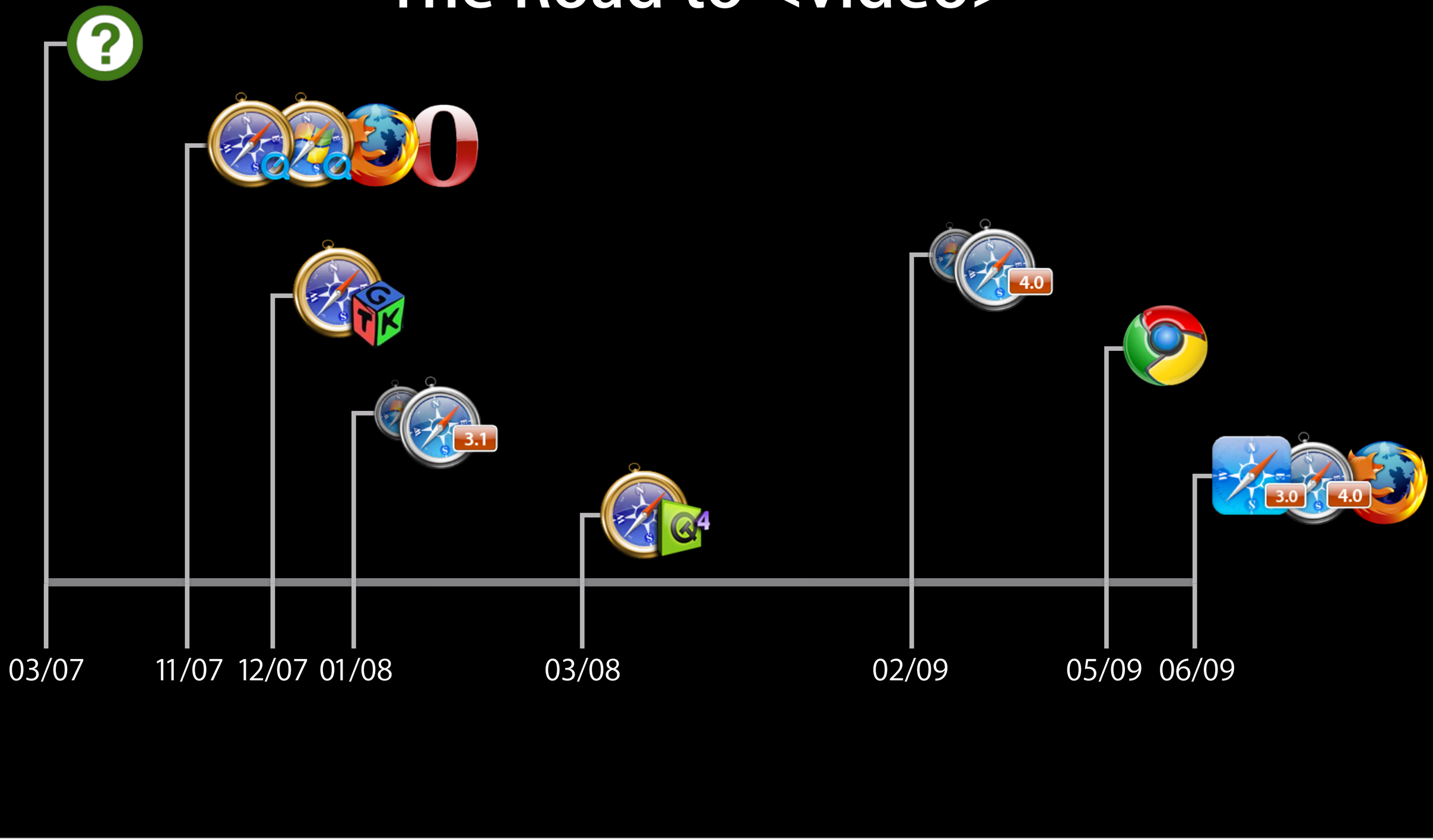
- Not all machines are created equal
- Media experience is core to the browser

Media should be a first class citizen on the Web.

HTML 5

WHATWG

The Road to <video>



Early Adopters

Demo

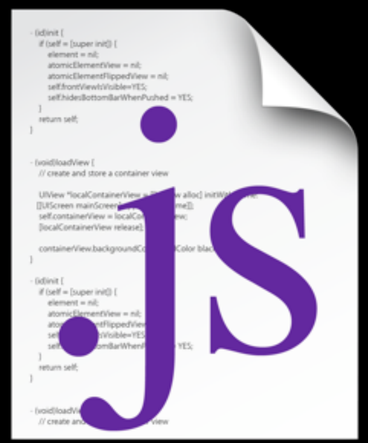
Wilco Video Player

Step 1—Setting the Stage

Basic <video> Integration

Web Page Structure

index.html



script.js

style.css

Stepping Into the Code

Step 1—Setting the Stage

What We Just Learned

<video> Attributes

- `controls`—display built-in controller
- `autoplay`—start playing automatically with enough data
- `poster`—custom placeholder image

More Attributes

- `autobuffer`—automatically load data without playing
- `loop`—constant repetition

Fallback

Supporting less capable browsers

```
<video src="movie.mp4" poster="poster.png" controls>
    <embed href="movie.mp4" src="poster.png">
</video>
```

Fallback

Related Session

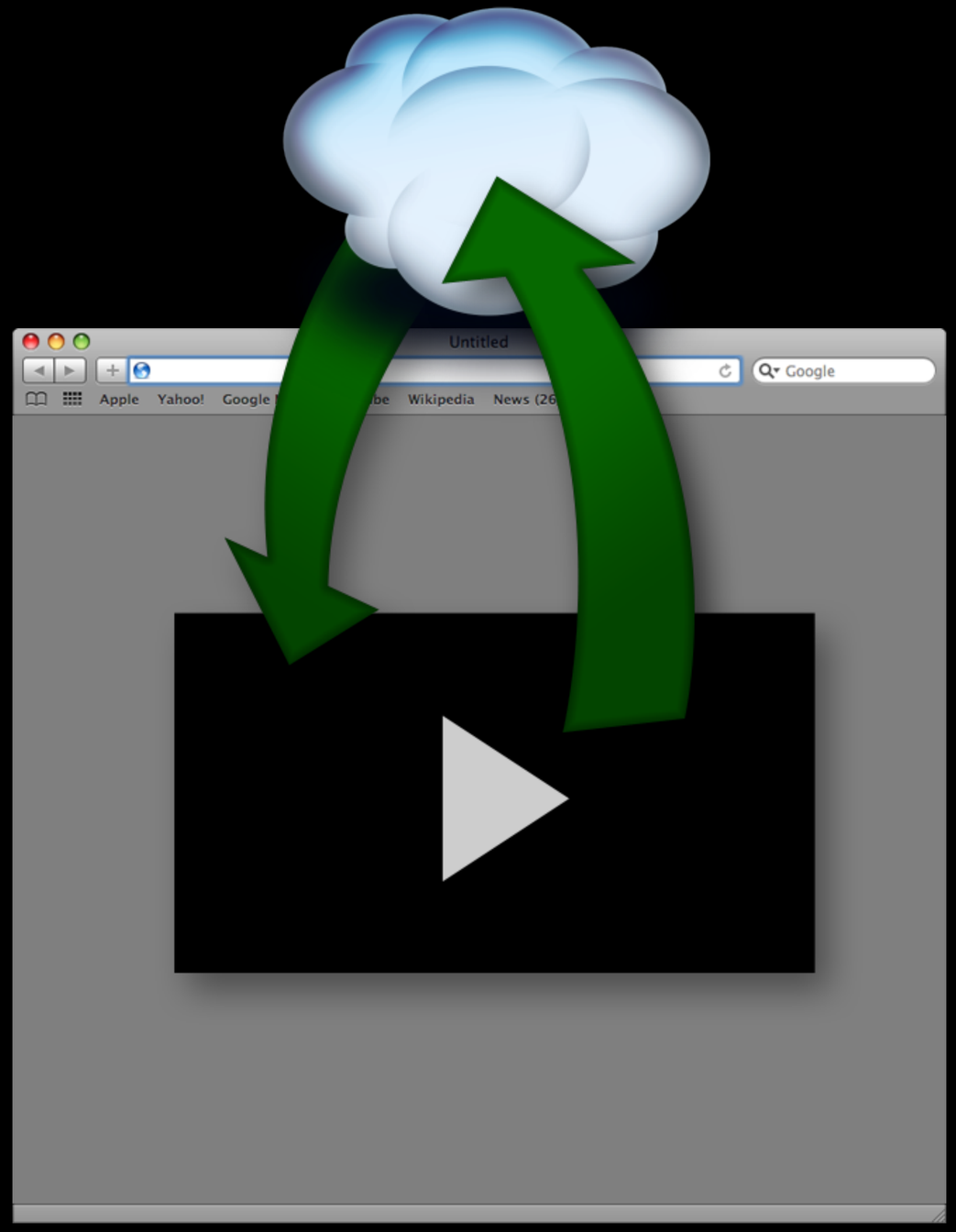
Building Compatible Websites Using Emerging Web Technologies	Russian Hill Thursday 5:00PM

Step 2—Simple Media Controls

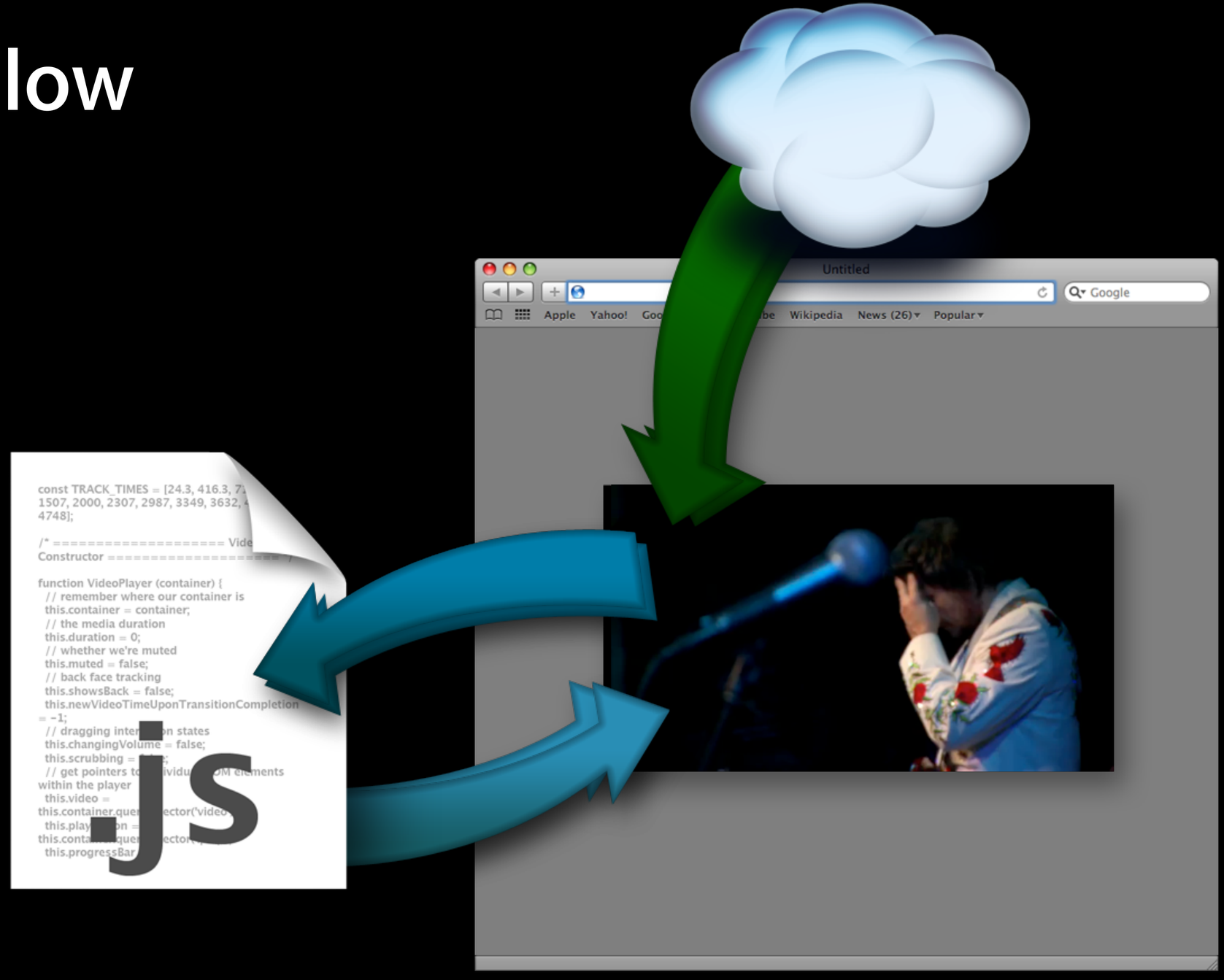
Basic Events and Media API Integration

PAUSE
VOL

Event Flow



Event Flow


const TRACK_TIMES = [24.3, 416.3, 7
1507, 2000, 2307, 2987, 3349, 3632,
4748];
/* ========== Vide
Constructor ========== */
function VideoPlayer (container) {
// remember where our container is
this.container = container;
// the media duration
this.duration = 0;
// whether we're muted
this.muted = false;
// back face tracking
this.showsBack = false;
this.newVideoTimeUponTransitionCompletion
= -1;
// dragging inter on states
this.changingVolume = false;
this.scrubbing =
// get pointers to ividu OM elements
within the player
this.video =
this.container.que ector('video
this.play on =
this.conta que ector(
this.progressBar
.js
Untitled
Google
Apple
Yahoo!
Wikipedia
News (26)
Popular

Stepping Into the Code

Step 2—Simple Media Controls

What We Just Learned

Media events

abort
canplay
canplaythrough
durationchange
emptied
ended
error
load
loadeddata
loadedmetadata
loadstart
pause
play
playing
progress
ratechange
seeked
seeking
stalled
suspend
timeupdate
volumechange
waiting

Media APIs

canPlayType()
play()
pause()
addCueRange()
removeCueRanges()
autobuffer
autoplay
buffered
controls
currentSrc
currentTime
defaultPlaybackRate
duration
ended
error
loop
muted
networkState
paused
playbackRate
played
readyState
seekable
seeking
src
startTime
volume

What We Just Learned

Coding techniques

- User action updates the media engine
- Media engine updates the UI via events
- Use CSS classes to drive the UI state

Step 3—Progress Bars

Time Control with Media APIs

PAUSE
VOL

Stepping Into the Code

Step 3—Progress Bars

What We Just Learned

Media events

```
abort
canplay
canplaythrough
durationchange
emptied
ended
error
load
loadeddata
loadedmetadata
loadstart
pause
play
playing
progress
ratechange
seeked
seeking
stalled
suspend
timeupdate
volumechange
waiting
```

Media APIs

```
canPlayType()
play()
pause()
addCueRange()
removeCueRanges()

autobuffer
autoplay
buffered
controls
currentSrc
currentTime
defaultPlaybackRate
duration
ended
error
loop
muted
networkState
paused
playbackRate
played
readyState
seekable
seeking
src
startTime
volume
```

What We Just Learned

Coding techniques

- Cache oft-requested media property
- Build CSS UI to rely on percentage values

Step 4—Scrubbing

Jumping Through Time

PAUSE
00:00:00
00:05:01
VOL

Stepping Into the Code

Step 4—Scrubbing

What We Just Learned

Coding techniques

- Query UI metrics in real time with `getBoundingClientRect()`
- Pause while scrubbing
  - Better performance
  - Better user experience

Step 5—CSS Transitions

Effortless Implicit UI Animations

Stepping Into the Code

Step 5—CSS Transitions

What We Just Learned

Coding techniques

- Trigger transitions declaratively based on states
- Let CSS handle complex presentation

Demo—Next Level

Advanced Wilco Video Player

Eric Carlson
Rich Media Team
Internet Technologies

Summary

Simple and powerful integration

- Just another HTML element
- Great built-in experience
- Style, transition and transform with CSS
- Monitor playback and timing with DOM events
- Control playback and timing with DOM APIs

Summary

Ready to deploy

- Supported on iPhone and desktop
- Supported or fallback in non-WebKit browsers

More Information

Vicki Murley
Web Technologies Evangelist
vicki@apple.com

Documentation
Safari Visual Effects Guide
Safari DOM Extensions Reference
http://developer.apple.com/safari/

Related Sessions

Building Compatible Websites Using Emerging Web Technologies	Russian Hill Thursday 5:00PM
Adding Innovative Styling and Animation with CSS Effects	Mission Tuesday 3:30PM

Labs

Media Playback in Web Applications Lab	OS X Lab C Thursday 10:30AM

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09